

Software Requirements Specification SRS

Help Me Im Stuck

Nikyar Karimi, Stefan Wesely, Haris Kurtagic

Document history

Version	Status	Date	Responsible person	Reason for change
0.01	not finished	21.11.2024	Nikyar Karimi, Stefan Wesely, Haris Kurtagic	project start, brainstorming und specifying requirements
1.00	finished	24.11.2024	Nikyar Karimi, Stefan Wesely, Haris Kurtagic	Defined requirements in rupps template, added use cases & diagram, specified product scope

Table of contents

1. INTRODUCTION	1
1.1. PURPOSE	1
1.1.1. <i>Background and goals of the project</i>	1
1.1.2. <i>Organizational embedding</i>	1
1.2. DEFINITIONS	1
1.3. PRODUCT SCOPE	2
1.3.1. <i>Must criteria</i>	2
1.3.2. <i>Should criteria</i>	3
1.3.3. <i>Could criteria</i>	4
1.3.4. <i>Won't criteria</i>	4
2. USE CASES	5
2.1. USE CASE DIAGRAM	5
2.2. USE CASES	6
2.2.1. <i>Create Ticket</i>	6
2.2.2. <i>Add Tag to Ticket</i>	6
2.2.3. <i>Add Comment to Ticket</i>	7
2.2.4. <i>Edit Department</i>	7
2.2.5. <i>Add Another Department to Ticket</i>	7
3. NON-FUNCTIONAL REQUIREMENTS	7
3.1. EASE OF USE	7
3.1.1. <i>Screen sizes</i>	7
3.2. RELIABILITY	8
3.2.1. <i>Concurrent user support</i>	8
3.3. SPEED	8
3.3.1. <i>Response time</i>	8

1. Introduction

1.1. Purpose

The purpose of this project is to develop a ticketing system that serves as a centralized solution for the management and tracking of requests within a company. This system is intended to support various departments such as support, but also other organizational units within the company, by enabling a structured and transparent processing of various types of requests. The purpose of the ticketing system is to increase the efficiency of internal communication, speed up the processing of inquiries and ensure that processes are recorded.

1.1.1. Background and goals of the project

The background of the project lies in the need to improve the efficiency and transparency of internal processes in companies, especially in departments that are regularly confronted with a high number of requests and tasks. Currently, many companies still have isolated systems or manual processes for managing requests, such as the disorganized sending of emails, which leads to misunderstandings, delays and a lack of traceability. Especially in departments such as IT support, legal and finance, a structured and traceable way of working is crucial to respond quickly and efficiently to concerns.

1.1.2. Organizational embedding

The ticketing system must be seamlessly integrated into the existing work processes and systems. A key aspect of the organizational embedding of the ticketing system is its development as a web application. This feature of the product ensures that the system can be seamlessly integrated into the existing infrastructure without the need to install special software on employees' end devices. As a web app, the ticketing system is accessible via the web browser, which enables a high degree of flexibility and user-friendly use across different end devices.

1.2. Definitions

The system is designed to have several essential components. These are necessary for the functionality of the software.

One of these is the presence of tickets. A ticket typically represents a problem or a request that requires a solution or answer. It is used to organize communication between requesters and supporters. As in the image below, the properties of a ticket are color coded in must (red) should (orange) and could (green). While the ticket must have a title, description, status, department(s) and a timeline, it should have comments and tags, and it could include images.

<i>Ticket</i>
title
description
status
department(s)
timeline
comments
tags
images

A user is the person who is using the application. A user must have a name and role(s). The user could have permissions.

<i>User</i>
name
role(s)
permissions

A department is a group of users who belong to a specific area or function within a company. This group typically shares similar tasks, responsibilities, and goals that focus on a particular aspect of the business. A department must have the property of name, users and roles. It should have tags and could have a description.

<i>Department</i>
name
users
roles
tags
description

A role is a set of permissions assigned to users within a system. It defines what actions a user can perform based on their position or function within an organization, like the admin or a supporter. Roles must contain a name, description and permissions.

<i>Role</i>
name
description
permissions

A tag is a keyword which is assigned to a ticket. These tags allow to filter and search for specific items based on the assigned keywords. Tags should have a name, abbreviation and style. They could have a description.

<i>Tag</i>
name
abbreviation
style
description

1.3. Product scope

1.3.1. Must criteria

- The system shall provide the users with the ability to log in to the application with valid credentials.
- The system shall have at least one administrator account.

- The system shall provide an administrator with the ability to create users.
- The system shall provide an administrator with the ability to delete users.
- The system shall provide an administrator with the ability to modify users.
- The system shall provide an administrator with the ability to assign roles to users.
- The system shall provide an administrator with the ability to add a user to a department.
- The system shall provide an administrator with the ability to remove a user from a department.
- The system shall have tickets with the defined must properties.
- The system shall prevent editing of the ticket title after creation.
- The system shall prevent editing of the title description after creation.
- The system shall provide a requester with the ability to add tickets.
- The system shall provide a requester with the ability to add a description to the ticket.
- The system shall provide a supporter with the ability to accept tickets.
- The system shall provide a supporter with the ability to decline tickets.
- The system shall provide a supporter with all the permissions a requester has.
- The system shall ensure the existence of a database.

1.3.2. Should criteria

- The system shall provide a requester with the ability to pull back their own tickets.
- The system shall prevent a requester from pulling back a ticket if it has already been accepted.
- The system shall provide a requester with the ability to add comments to their own tickets.
- The system shall provide a supporter with the ability to add comments to all tickets.
- The system shall prevent the editing of ticket comments after they have been created.
- The system shall prevent a supporter from accepting tickets that they have created themselves.
- The system shall provide a supporter with the ability to return an accepted ticket back to the queue.

1.3.3. Could criteria

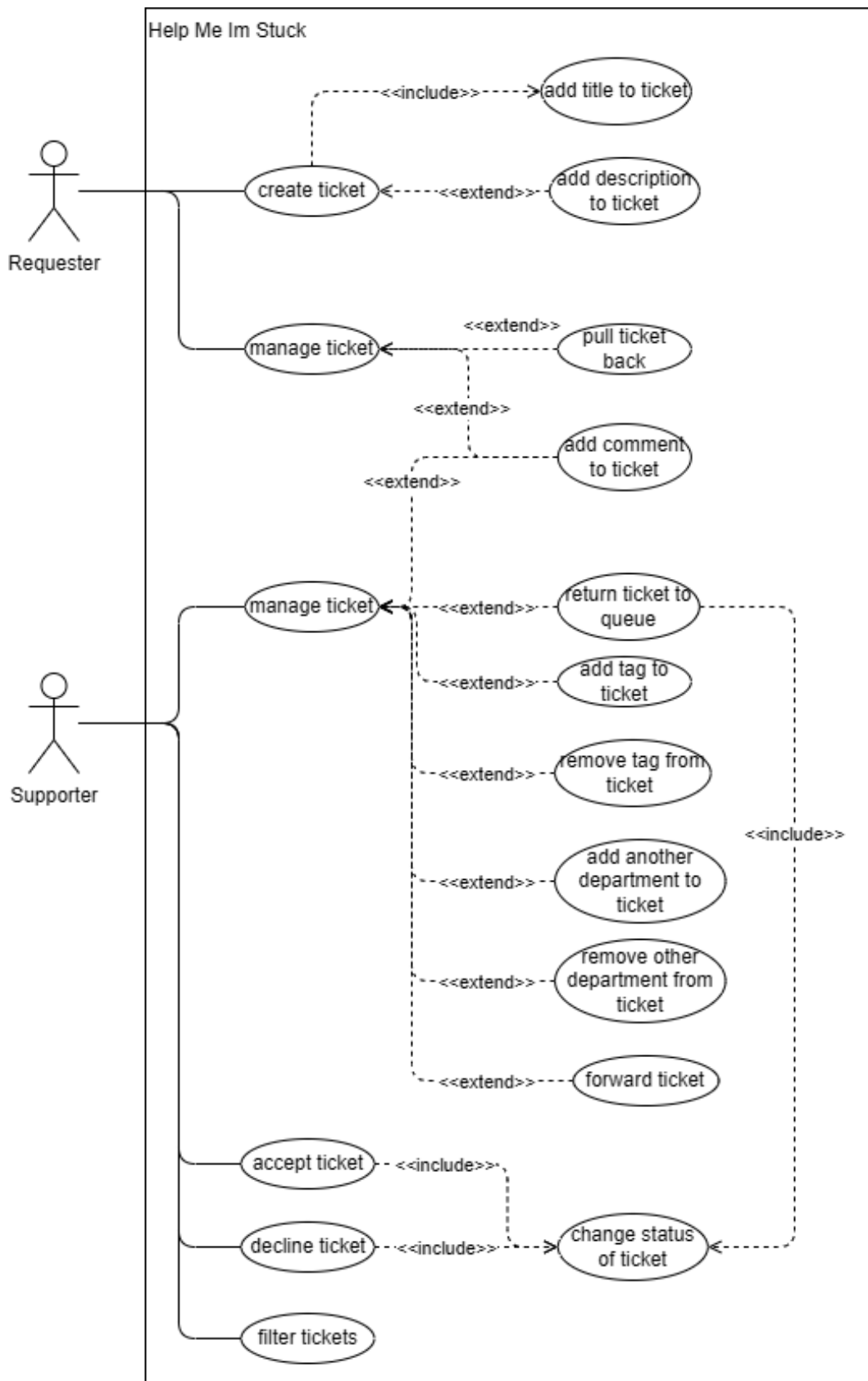
- The system will provide a supporter with the ability to add another department to a ticket.
- The system will require all departments to close a ticket before it can be marked as closed.
- The system will provide a supporter with the ability to forward a ticket to a different department.
- The system will provide a requester with the ability to add an image to a ticket.
- The system will provide an administrator with the ability to create roles.
- The system will provide an administrator with the ability to modify roles.
- The system will provide an administrator with the ability to delete roles.
- The system will provide an administrator with the ability to create new departments.
- The system will provide an administrator with the ability to modify departments.
- The system will provide an administrator with the ability to delete departments.
- The system will ensure that tags are only visible in their respective department.
- The system will provide statistics regarding the ticket processing.
- The system will provide departments with the ability to see their statistics.

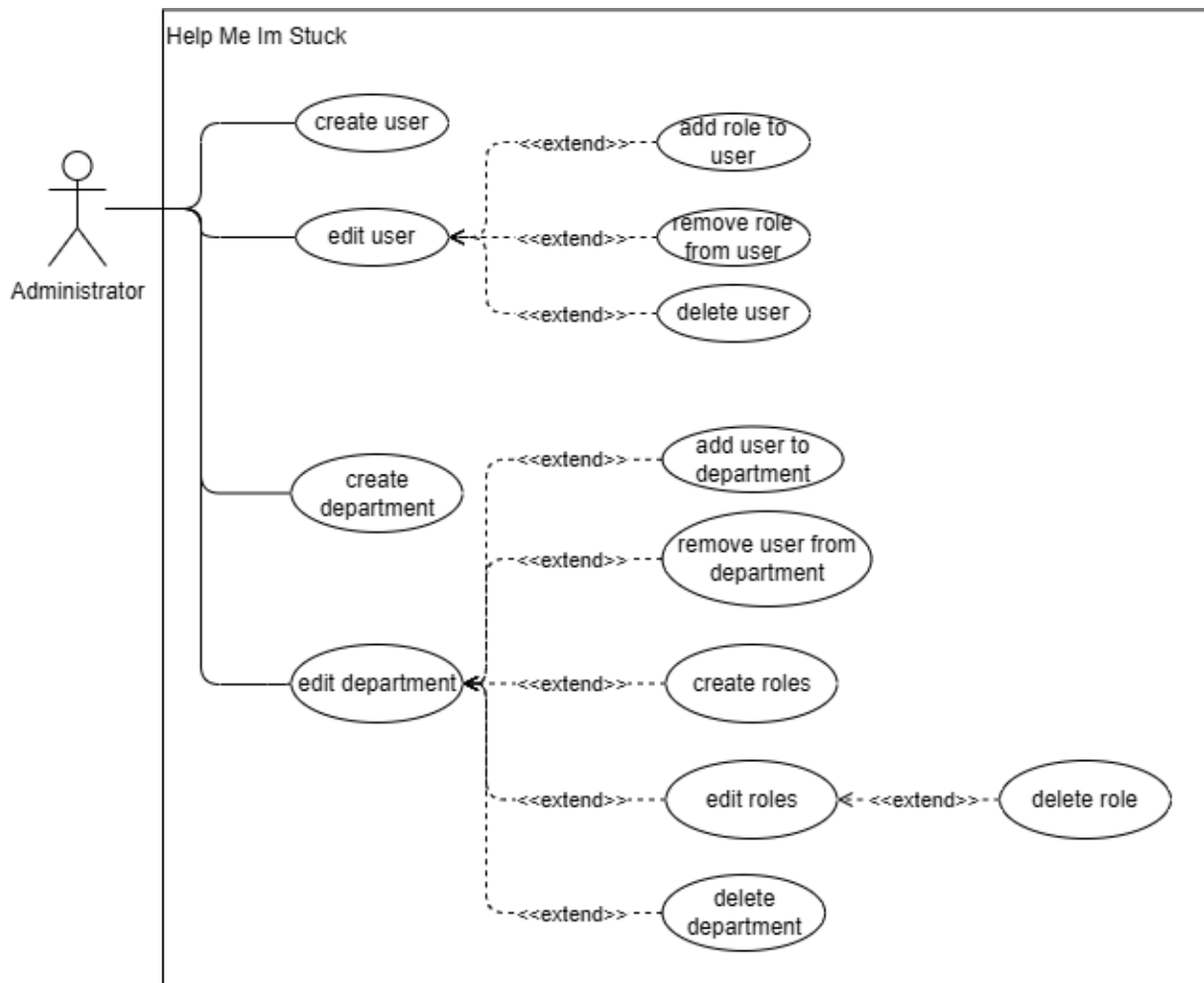
1.3.4. Won't criteria

- The system won't allow guest users.
- The system won't require email verification.
- The system won't allow tags to have a correlation with permissions.
- The system won't support any platform other than the web app.

2. Use Cases

2.1. Use Case Diagram





2.2. Use Cases

2.2.1. Create Ticket

Use case	Create Ticket
Actor	User
Description	A user creates a new ticket to report an issue or request assistance.
Stimulus	The user starts the ticket creation process.
Response	The system prompts the user to add a title and description to the ticket.
Comment	

2.2.2. Add Tag to Ticket

Use case	Add Tag to Ticket
Actor	Supporter
Description	A supporter assigns a tag to a ticket for better categorization or prioritization.
Stimulus	The supporter selects a ticket from the queue and chooses the "add tag" option.
Response	The system displays a list of predefined tags. Once selected, the tag is associated with the ticket and displayed in the ticket's metadata.
Comment	

2.2.3. Add Comment to Ticket

Use case	Add Comment to Ticket
Actor	Supporter
Description	A supporter adds a comment to a ticket to provide the ticket creator with information or to get further information from the ticket creator.
Stimulus	The supporter selects a previously accepted ticket and chooses the "add comment" option.
Response	The system displays a text box to enter the comment into. Once filled and confirmed, the comment is associated with the ticket and displayed in the ticket's timeline.
Comment	The supporter must accept the ticket before being able to add a comment.

2.2.4. Edit Department

Use case	Edit Department
Actor	Administrator
Description	An admin modifies an existing department's details, such as adding or removing users.
Stimulus	The admin selects a department to edit.
Response	The system updates the department's details based on the admin's inputs.
Comment	Ensures departments are correctly managed to reflect organizational changes.

2.2.5. Add Another Department to Ticket

Use case	Add Another Department to Ticket
Actor	Supporter
Description	A supporter assigns an additional department to a ticket when the issue requires input or action from multiple departments.
Stimulus	The supporter selects the "add another department" option from the ticket's menu and chooses the department to involve.
Response	The system links the ticket to the additional department, places the ticket in the queue of the added department and updates the ticket's metadata to reflect the involvement of multiple departments.
Comment	The supporter must accept the ticket before being able to add another department.

3. Non-functional Requirements**3.1. Ease of Use****3.1.1. Screen sizes**

The frontend of the app shall support two different screen sizes

3.2. *Reliability*

3.2.1. **Concurrent user support**

The app should be able to handle at least 5 concurrent users.

3.3. *Speed*

3.3.1. **Response time**

The app should respond to user actions within 5 seconds.